

REMARKS

This is in response to the office Action dated March 25, 2008, in which claims 1-20 continue to be rejected. The above amendments are clarifying amendments not made in view of any prior art reference.

I. BASIC MISUNDERSTANDING/MISDESCRIPTION IN OFFICE ACTION

The claim rejections and the various comments and analysis provided by the Examiner in the Office Action reflect a basic, underlying misunderstanding or misdescription of expressly-recited claim elements.

The analysis provided in the Office Action fails to consider, within the context of the other claim elements, that:

- multiple instances of the same peripheral device are instantiated within a single chip design;
- at least two of the instances within the same chip have different configurations from one another; and
- the different configurations are selected at compile time without modification of the hardware description of the peripheral device.

As a result, the Office Action attempts to combine references that fail to meet the plain language of Applicant's claims.

The following arguments are divided into two sections. The first section provides brief replies to the Examiner's responses to Applicant's arguments provided in pages 2-6 of the Office Action. The second addresses the specific claim rejections.

II. APPLICANT'S BRIEF REPLIES TO EXAMINER'S RESPONSES

Examiner's Response, Section 5, pages 2-3

The Examiner's first point is correct that Bowen teaches using a configuration (without modification to that particular configuration) multiple times. But Bowen does not disclose multiple instances of a peripheral device on the same IC with different configurations without modification to the hardware description of the peripheral device.

Applicant is confused by the second point relating to Bowen paragraph [0241]. In this part of the patent, Bowen is talking about the preferred embodiment written using JAVA, C, & C++ languages and uses object oriented programming (OOP) methodology. Paragraph [0241] describes some of the features of OOP, where OOP components are reusable software modules that can be used together. Applicant does not see how this relates to or teaches tying a single code block to hardware or how this relates to Applicant's claims.

Examiner's Response, Section 5, page 3

Duboc et al. does not disclose using the same building block more than once with different input from the user in the same IC, at compile time. This point is described in technical detail below in response to the rejection.

Examiner's Response, Section 5, top of page 4

Duboc never says different instantiations started from the same reusable "template".

Examiner says "customizable circuit block to be included in the custom (i.e. having different instantiations) DSP integrated circuits (col. 3, lines 7-15)."

This is the wrong use of the term "custom". The DSP IC is custom from other DSP ICs, but does not include multiple instances of a separately customized DSPs (different options) on the same IC.

An embodiment of the present application looks at one IC, not multiple ICs, though two or more ICs could be simulated/compiled at the same time in other embodiments.

Looking at Duboc more closely, Figure 3 does not show the same parametrizable block twice with different parameters/configurations. Figure 3 shows Y ROM & P ROM (Y RAM & P RAM) implying that they are different RTL blocks. For example, there is no an instance identifier in the template of FIG. 6 (so as to independently configure different instances in different ways).

An embodiment of the present application would allow Y-ROM & P ROM to be the same RTL description (say Z ROM), with each instance having different parameters/configuration options.

Duboc inputs a template file (.TPL, .STPL) and scripts (.HPC, .HTF) (column 8, 51-62) to output the RTL description (VHDL/Verilog). The present application uses, for example, the same RTL file as input and output. The parameters and strap pins work with the way the RTL is written to allow the different configurations for multiple uses.

Applicant cannot find anywhere that Duboc where says the same RTL (Verilog/VHDL) output (module name) is used twice in the same custom DSP IC.

Examiner's Response, Section 5, page 5

The Examiner states incorrectly that:

"Duboc's GUI system let user to select the options to have a customizable circuit block to be included in the custom (i.e. having different instantiations) DSP integrated circuits (col. 3, lines 7-15)."

In claim 1, for example, Applicant recites one IC. In contrast, Duboc wants to quickly generate multiple "custom" ICs (where one IC is different than the other IC. Different problem are getting solved.

Further, the Examiner states, incorrectly, that Duboc teaches: "method of selecting between the options at compile time for each instance of the peripheral device such that at least two of the instances have different configurations from one another".

Applicant does not agree. Duboc teaches a GUI method of selecting options and outputting RTL. It does not teach that two or more of these outputs have the same RTL module name (from the same template input) but have different configurations at the time the design RTL (IC RTL) is compiled into a simulator or synthesis tool.

Examiner's Response, Section 5, page 6

Regarding Applicant's remarks concerning the strap pins of Yu, Applicant agrees that terms in the specification should not be incorporated into the claims.

Claims 4 and 12 are amended to further define the method steps not taught by Yu.

Claim 4, for example, uses the strap pin to select between options at compile time (e.g., by tying an input to a logical level), and not as described by Yu, to effect (fix) power distribution issues a chip can have such as the power level (Vdd=5.0) dropping 4.0V.

Yu is irrelevant to claims 4 and 12.

III. CLAIM REJECTIONS UNDER §103 BASED ON BOWEN AND DUBOC ET AL.

Claims 1-3, 5-11 and 13-20 were rejected under §103(a) as being allegedly unpatentable over U.S. Publication No. 2002/0100029 to Bowen in view of U.S. Patent No. 6,425,116 to Duboc et al.

A. **Bowen**

In general, Applicant does not follow why The examiner refers to Bowen paragraphs 0041, 009, 0036, 0138. They have some of the same terms like compile and configuration, but different things are going on.

More specifically, the Office Action incorrectly suggests that Bowen discloses “configuring a function block to instantiate multiple instances of the peripheral device within the single chip, . . . the hardware description having options associated with different configurations of the peripheral device . . . wherein the options are selected without modification to the hardware description. . . .”

1. Options are Not Selectable Without Modification To The Hardware Description

In Bowen, Paragraph [0036] states, “a hardware compiler for producing from those parts of the specification partitioned to hardware a register transfer level description for configuring configurable logic resources.”

The register transfer level description configures the configurable logic resources according to the input provided to the hardware compiler. The input provided to the hardware compiler contains a pre-defined configuration that is used for configuring the configurable logic resources, such as during synthesis at step 214 in Figure 2. The input itself is not configurable at compile time. Rather it is predetermined by the C code. Thus, a configuration change would require a change to the hardware description.

The Examiner may be suggesting that the C-like input description is equivalent to the claimed "hardware description" (e.g. an RTL description). However, Bowen shows you can create multiple RTL descriptions from the C-like input description (it is reusable). Whereas, claim 1 allows creation of multiple configurations ("uses") from the same hardware (e.g., RTL) description.

The Examiner refers to Bowen, paragraphs [0040-0041]. Paragraph [0040] talks about taking a behavioral description of the target electronic system and automatically partitioning the required functionality between hardware and software while being able to vary the parameters (size or power) of the hardware and/or software. Varying the parameters of the hardware or software effects the output (the desired netlist or register transfer level (RTL) description). That netlist or register transfer level description is then used in their FPGA or ASIC [0042]. It is implied that this output is used "as-is" because there is no teaching or suggestion of further modifications of that netlist or register transfer level description or multiple instances ("uses") of that netlist or register transfer level description in the same FPGA or ASIC.

Bowen Figure 2 shows a C-like input description producing an output "RTL description" (box 212). The "width adjustment" blocks [206] are before block 212 in the flow and therefore before the RTL description is created. This RTL description is then targeted to an FPGA/ASIC [0144, 0145]. Paragraph [0127] talks about estimating the speed of the hardware.

Thus, it is incorrect to state that Bowen does disclose options selected at compile time "without modification to the hardware description".

Further, the discussion of Figure 2 does not talk about using this RTL description multiple times in the same chip (FPGA/ASIC). In contrast, the present application discusses applying the configuration options to the RTL description to have the same RTL description used multiple times on the same chip but each instance (use) can (but doesn't have to) perform/act/behave differently. In claim 1, at least two instances have different configurations from one another.

2. Bowen Does Not Disclose Selecting Between the Options at Compile Time for Each Instantiation of the Peripheral Device

The Office Action acknowledges that Bowen does not disclose “selecting between the options at compile time for each instance of the peripheral device such that at least two of the instances have different configurations from one another,” as recited in claim 1.

In one non-limiting example of the present application, RTL is coded in a way so that different configurations can be used in a single chip (where the configuration options are selectable at compile time) without modifying the RTL for each instance.

The Office Action suggests these limitations would be obvious in view of Duboc.

C. **Duboc et al.**

1. Duboc et al. Fails to Disclose That Two Different Instantiations Can Have Two Different Configurations Selectable at Compile Time

The Office Action incorrectly states Duboc et al. shows “selecting between the options at compile time for each instance of the peripheral device such that at least two of the instances have different configurations from one another,” citing Duboc, col. 8, lines 33-39; col. 10, lines 31-34.

Duboc et al. generally discloses a way to take existing building blocks and user input to make RTL, reduce to gates, and make an integrated circuit. Duboc et al. does not disclose using the same building block more than once with different input from the user in the same IC.

Column 8, lines 33-39 mentions using the GUI input to initiate generation of the IC design via selection of a “compile option” from the GUI window, resulting in the execution of a script engine that verifies parameters input by a user.

In Duboc et al., the user configures the IC once. In the example described in column 6, lines 47-57, they are making one custom DSP IC. Looking at the options of Table 1, the user picks one Data Y ROM size to make this one custom DSP IC. The user does not run the GUI (or a template) twice to have two or more custom DSP configurations (different DSP instances) on one IC. Rather, the user runs through the GUI once and it configures the IC.

For example, there is no an instance identifier in FIG. 6 (so as to independently configure different instances in different ways). The Y RAM and X RAM are not two different of the same RAM instances, but are tied together with the same configuration (their address space is divided). “The Y data space, which is configurable by the user, can occupy the top 2, 4, 8 or 16K of the data space, with the remaining data addresses mapped to the X data space.” (Col. 6, lines 47-57; see also, col. 9, lines 43-47). They are not separately configurable instances of the same peripheral device.

In the present application, the same building block (peripheral) can be used twice in the same IC and have a different configuration. So, for example, the user could configure the Data Y ROM to be 2K in one custom DSP instance on the IC and the Data Y ROM to be 4K in another/different custom DSP instance on the same IC.

Duboc et al. do not disclose that two different instances of the same peripheral device can have two different configurations selectable at compile time. Referring to the language of claim 1, Duboc et al. do not disclose “configuring a function block to instantiate multiple instances of the peripheral device within a single chip design, the hardware description of the peripheral device having options associated with different configurations of the peripheral device; [and] selecting between the options at compile time for each instance of the peripheral device such that at least two of the instances have different configurations from one another, wherein the options are selected without modification to the hardware description.”

In fact, Duboc et al. do not suggest that such a feature would be desirable or how such a feature could be accomplished. The disclosed GUI does not provide that structure or functionality.

D. Combination of Bowen and Duboc et al.

Lacking such a disclosure, the proposed combination of Bowen and Duboc et al. therefore fails to teach or suggest multiple instances of a peripheral device on the same IC with different configurations, where the same function block is used to instantiate a hardware description with options associated with the different configurations of the peripheral device.

The proposed combination also fails to disclose a step of selecting between the options at compile time for each instance of the peripheral device without modification to the hardware description.

Further, there is nothing in either Bowen or Duboc et al. that would make it obvious for a skilled person to try.

E. Dependent Claims 2 and 3 (For Example)

Claim 2 adds that the step of selecting comprises “passing a parameter value to the function block at compile time for each instance of the hardware peripheral.”

Applicant does not understand why the Examiner cited Bowen paragraph [0111] or how it applies to claim 2. This paragraph refers to the user defining the scheduling decisions (operations happen on certain clock cycles) to the compiler. This is the compiler taking the C code (or RTL language) and creating the hardware description to put into the FPGA. See para [0105].

Claims 3 describes that the configuration options are peripheral design functions, peripheral design pin widths, or peripheral design interface pin outs.

Again, neither Bowen nor Duboc et al. disclose a method in which a parameter value can be passed to a function block that is configured to instantiate each instance of a hardware peripheral, where the different instances can have different configurations (per claim 1).

The Office Action cites Bowen paragraph [0037], which states that “The system can include a width adjuster for setting and using a desired data word size, and this can be done at several points in the desired process as necessary.”

But in Bowen, the same width adjustment would be applied to every instance in

the design. Paragraph [0114] describes that at 206a and 206b, width adjustment is carried out by ANDing bits with a bit mask. This would apply to every instance of the block/code in the design.

In the present application, the functional block can be configured to use a different width for each instance of the block in the design, for example.

In addition, one or more other dependent claims add further elements that are neither taught nor suggested by Bowen and/or Duboc et al.

F. Claim 7

In respect to independent claim 7, the Office Action refers to paragraphs [0009], [0031] and [0241] of Bowen. Paragraph [0241] simply states that the OOP components (reusable software modules) are accessed at run-time.

As discussed above, Bowen does not disclose configuring a function block to instantiate a hardware description with options at compile time and instantiating multiple instances of the peripheral device on the integrated circuit by programmatically selecting between the options at compile time for each instance of the peripheral device.

As described above, Duboc et al. also fails to disclose these elements.

Claim 7 and its respective dependent claims are therefore not anticipated by or obvious over Bowen in view of Duboc et al. for similar reasons as were discussed above with respect to claim 1.

G. Claim 16

Similarly, independent claim 16 and its dependent claims are not anticipated by or obvious over Bowen in view of Duboc et al. for the similar reasons as were discussed above with respect to claims 1 and 7.

IV. CLAIM REJECTIONS UNDER §103 BASED ON BOWEN, DUBOC ET AL. AND YU ET AL.

Claims 4 and 12 were rejected under §103 as being allegedly being unpatentable over Bowen in view of Duboc et al. and Yu et al., U.S. Patent No. 6,829,754.

The combination of Bowen, Duboc et al. and Yu et al. does not teach or make obvious the inventions recited in claims 4 and 12 for similar reasons as discussed above with respect to the independent claims.

In addition, page 13 of the Office Action suggests "one of ordinary skill in the art would be motivated to strap the power or ground pins so that the power related problems can be avoid[ed]".

The strap pins discussed in claims 4 and 12 are not for power related problems. They are to tie peripheral input pins logic level high or logic level low (based on the particular configuration of that instance) to make a functional decision on different features supported in a design. For example, processors and computer systems have the idea of big endian data format and little endian data format. These formats specify how bytes are organized in the memory. A peripheral can have a static strap pin tied to ground (low) if little endian data format should be used or to power (high) if big endian data format is used. The pin is tied high or low during configuration.

The comment the examiner makes about power related problems and the citation to Yu column 11, lines 2-5 relate to current flow or power through an actual piece of metal and generating a warning if the physical width of the strap is smaller than the power pin to which it connects. This has nothing to do with configuring a functional block to tie a strap pin to power or ground for implementing a particular configuration of a peripheral device.

Accordingly, Applicants respectfully request that the objections of these claims under §103 be withdrawn.

The Director is authorized to charge any fee deficiency required by this paper or credit any overpayment to Deposit Account No. 12-2252.

Respectfully submitted,

WESTMAN, CHAMPLIN & KELLY, P.A.

By: /David D. Brush/
David D. Brush, Reg. No. 34,557
900 Second Avenue South, Suite 1400
Minneapolis, Minnesota 55402-3319
Phone: (612) 334-3222 Fax: (612) 334-3312